

SEVENTH QUIZ

You have 15 minutes from the start of class to complete this quiz. Read the problems with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

Brief reminders: `odd?` returns true if its input is an odd number; `zero?` returns true if its input equals zero; `sub1` subtracts one from its argument; the function `quicksort` behaves as follows:

```
;; quicksort: list-of-X (X X -> boolean) -> list-of-X
;; Return the input list sorted according to the second arg., a comparison function.
```

Problem 1 (6 points)

What is the value of each of the following expressions?

- (a) `(map sub1 (list 31 21 11 1))`
- (b) `(map (lambda (n) (* n 10)) (list 2 4 6 8 10))`
- (c) `(filter odd? (list (+ 3 4) (* 5 2) (/ 350 10) (sub1 1)))`
- (d) `(filter (lambda (n) (or (zero? n) (= 1 n))) (list 0 1 2 3 2 1 0))`
- (e) `(foldr + 0 (list 10 20 30 40))`
- (f) `(foldr (lambda (a b) (and a b)) true (list true true false true))`
- (g) `(build-list 4 (lambda (n) n))`
- (h) `(build-list 4 sub1)`

Problem 2 (6 points)

Suppose we have a list called BL of books defined as follows:

```
(define-struct book (title author genre price sold instock))
```

where `title` and `author` are strings, `genre` is a string (e.g., "cookbook" or "humor") representing the category of the book, `price` is a number representing the price of one copy, `sold` is the number of copies sold, and `instock` is the number of copies in stock.

For each of the following expressions, describe in one clear and precise English phrase what value it returns. Don't just say, "It does a foldr of plus and zero and ..."; give a description of what the expression *means*, something you could put in a software catalog so that a prospective buyer could find what he or she wanted. Use real-world terms, not program syntax terms: Say something like, "a list of the authors whose books earned over \$1,000,000," not "books whose book-sold field is greater than 1000."

- (a) `(filter (lambda (B) (string=? (book-author B) "Arthur Conan Doyle"))
 (filter (lambda (B) (> (book-instock B) 0)) BL))`

(b) `(first (map book-title
 (quicksort BL (lambda (B1 B2) (> (book-sold B1) (book-sold B2))))))`

(c) `(filter (lambda (B) (member (book-genre B) (list "Law" "Politics" "Government"))) BL)`

Problem 3 (8 points)

(a) (2 points) Complete the definition of the following function:

```
;; book-earnings: book -> number
;; Return the total amount of money received from sales of the book
(define book-earnings
  (lambda (B)
```

(b) (6 points) Using `map`, `filter`, and/or `foldr` as necessary, and functions defined earlier where possible, define the following function without using explicit recursion. [Hint: You may use `local` to define intermediate results.]

```
;; earnings-for-genre: list-of-book string -> number
;; Return the total amount received from sales of books of the specified genre
(define earnings-for-genre
  (lambda (L g)
```